

unfolding Color Code の誤り耐性 その3

前回に引き続き、Color Code の unfolding 操作について、符号距離 3 のときの誤り耐性を調べてみた。今回は stim で符号距離 3 の場合を実装し、エラー伝播、測定エラーがあるような状況でも実効的な符号距離が 3 になっていることを確かめた。

1. Unfolding 過程のエラー検出

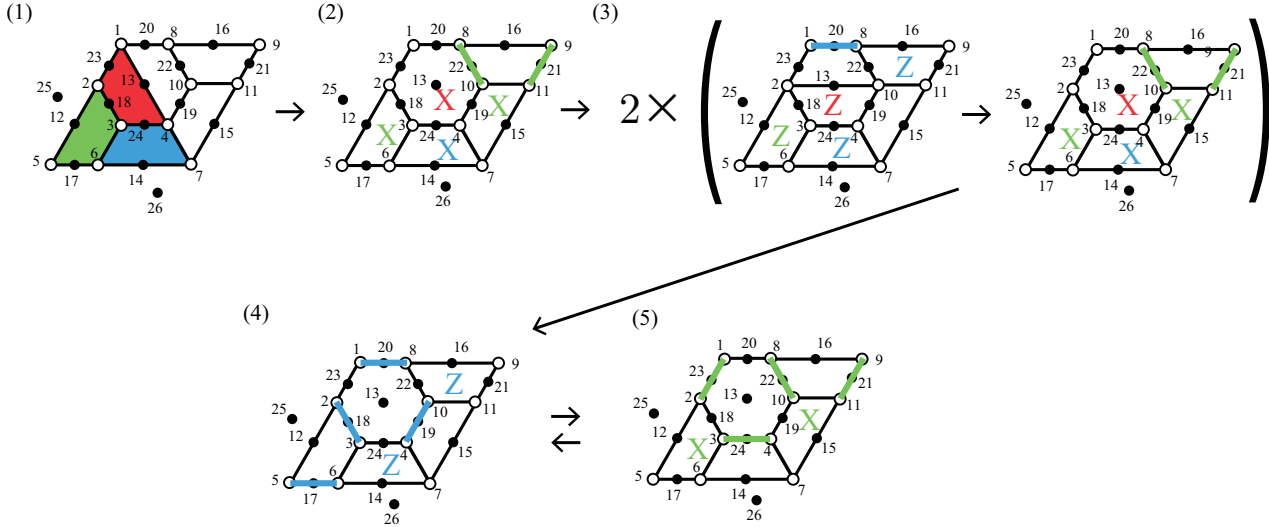


Fig. 1

前回の資料 (unfolding_color_code_3.pdf) からの変更点をまず解説する。最終的なプロトコルを Fig.1 に示す。白丸は data qubit、黒丸はその周りの data qubit に対してシンドローム測定をする ancilla qubit である。(1) について、ここは前回と変わっていない、右上の部分に unfolding するための qubit を $|0\rangle$ で用意する。qubit 25, 26 は後のエラー検出のために欠かせない flag qubit である。(2),(3) は前回と大きく異なる。まず (2) から、前は unfolding の始めは Z stabilizer でシンドローム測定をしていたが、それを X stabilizer からに変更した。また、ancilla 12 に関する 4-weight X stabilizer 測定については flag qubit を用いて 2 個の X エラーが伝播したかどうかを flag qubit 25 を用いて監視しておく（要はシンドローム測定の 2 個目と 3 個目の CNOT の間で ancilla 12 に X エラーが起きたかどうかを監視）。Ancilla 14 についても、同様に flag qubit 26 を使って 2 個の X エラーが伝播したかどうかを監視しておく。次に (3) は前回とは異なる形の Z Stabilizer 測定を行うようにし、変換途中の Z Stabilizer と X Stabilizer 測定を 2 回繰り返すように設計されている（ここでは、2 回繰り返すと書いているが、本質的には (2) の形の X stabilizer 測定を符号距離回だけ行うことがエラー検出するうえで必要だということ。詳細は後で説明。）。また、Z stabilizer 測定では ancilla 12 について 2 個の Z エラーが伝播したかどうかを監視、X stabilizer 測定は (2) と全く同じ。(4)(5) は前回と同じ。

それでは、(2) の形の X stabilizer 測定を符号距離回だけ行わなければならない理由を説明する。それは、ancilla 14 を使う青色 X stabilizer を捨てるからである。具体的に説明すると、例えば、(2) のシンドローム測定の CNOT で 2 qubit depolarizing error が ancilla 14 と data qubit 7 に確率 p で発生したとすると、そのエラーの種類が確率 $1/15$ で ancilla 14 に Z ゲート、data qubit 7 に Z ゲートだった場合、そのようなエラーが起きる確率は $p/15$ である。また、このエラーは (2) のシンドローム値を反転させない、かつ red boundary に存在する X logical operator と反可換である。そのため、このエラーを検出するためにもう一回青色 X stabilizer のシンドローム値を得ないといけない（これは、(3) の 1 回目の X stabilizer 測定に対応）。しかし、またここで ancilla 14 に measurement error が確率 p で発生したとすると、(2) で発生したエラーはまだ検出できない。また、ここまでのエラーは確率 $p^2/15$ で起きるので、符号距離 3 を担保するためにはもう一回青色 X stabilizer のシンドローム値を得ないといけない（これは、(3) の 2 回目の X stabilizer 測定に対応）。この 3 回目の測定でも measurement error が発生する確率は $p^3/15$ であ

るので、これは検出できなくて良い ($p^3/15$ が起きる確率は data qubit 1,2,5 にエラーが起きる確率 p^3 より小さい)。ということで、 p^{d-1} までの確率で起こるエラーを取り除くためには d 回同じことをしないとけない。

次に flag qubit を用いる理由について説明する。例えば (2) の ancilla 12 に注目すると、これを用いるシンドローム測定によって伝搬する 2 つの X エラーは、(i) CNOT を qubit 2,3,5,6 の順番でかければ、5,6 に伝播、(ii) CNOT を qubit 3,6,2,5 の順番でかければ、2,5 に伝播する (ここでは、stabilizer を法として合同なものは無視)。 (i) が起き、data qubit 7 エラーが起きると、実効的な符号距離は 2、(ii) が起き、data qubit 1 に X エラーが起きると実効的な符号距離は 2 なので、どの順番で CNOT をかけても符号距離が 2 になってしまう。そのため、flag qubit で 2 つの X エラーの伝播を監視する必要がある。他の flag qubit も同じ。

flag qubit を用いない weight が 4 以上の stabilizer については厳密に順番が決まっている。4-weight は前回紹介した通り、6-weight (今回は赤 X stabilizer) は例えば qubit 2,8,1,3,4,10 の順番である。これは総当たりで調べた。

2. 検証

検証は google の [1] の appendix C の 2 に載っていた MaxSAT を解く形で行った (中身は全く理解していない)。この検証で、Z メモリ、X メモリの実効的な符号距離がどちらも 3 になった。ただし、検証は Color Code のシンドローム測定でエラーが起こらないという仮定で行った。

REFERENCES

- [1] Nathan Lacroix et al. “Scaling and Logic in the Color Code on a Superconducting Quantum Processor”, arXiv preprint. eprint: 2412.14256. URL: <https://arxiv.org/abs/2412.14256>.